CHAPITRE 3

CONCEPTION DU SYSTEME

Introduction :

Comme nous l'avons présenté dans le chapitre 1, l'algorithme de classification hiérarchique ascendante nommé CAH est un algorithme de classification non supervisée de données, ce qui signifie qu'on ne fournit qu'une base d'entrée et c'est le système qui doit déterminer ses sorties. Dans ce chapitre, nous allons présenter les différentes étapes de conception, afin d'atteindre notre objectif en l'occurrence la classification des attaques réseaux pour la détection des intrusions.

Ce chapitre consistera donc en une problématique du système ensuite la démarche adoptée pour la conception puis nous y allons expliquer la phase de normalisation. Nous donnons en récapitulation un organigramme général décrivant le fonctionnent du système.

1. Présentation :

Dans de nombreuses approches de clustering, on s'intéresse à l'algorithme de classification ascendante hiérarchique, Dans le cas de CAH, à partir des éléments, on forme de petites classes ne comprenant que des individus très semblables, puis à partir de celles-ci, on construit des classes de moins en moins homogènes, jusqu'à obtenir la classe toute entière.

Dans un système de détection d'intrusions idéal, un classifieur apprend à partir d'un nombre suffisant de données qu'elles soient normales aussi bien qu'anormales. Puis, ce programme peut être utilisé pour classifier les données non vues dans ces deux catégories : normales ou anormales**.**

2. Problématique :

Les techniques de classification hiérarchique peuvent s'avérer particulièrement utiles pour rechercher les configurations de structure qui sont intéressantes. Dans le contexte de détection d'intrusions, ces configurations peuvent être associées à des attaques courantes, ou peuvent être des configurations de données normales ou des déviations aux schémas classiques qui ont été détectées.

Afin d'atteindre notre objectif, nous proposons une architecture appliquant l'algorithme de classification ascendante hiérarchique sur les paquets circulant dans un réseau local pour détecter les éventuels paquets frauduleux d'entre eux.

3. Démarche :

Notre système se concentre sur la détection d'anomalies pouvant surgir au sein d'un réseau local. Nous avons appliqué l'algorithme de CAH sur un ensemble standard de données kddcup99 [45] pour obtenir l'hiérarchie des connexions de cette base de données.

Le trafic réseau peut être capturé en utilisant des utilitaires de capture de paquets, puis stocké comme une collection d'enregistrements dans un fichier pcap.

L'algorithme CAH permet de représenter chaque point de données, soit une attaque ou une connexion normale.

4. Etapes de conception

La conception de notre système consiste respectivement en :

Une phase de normalisation de données (les données de la base d'apprentissage et les données de capture) et une phase d'explication détaillée sur l'algorithme CAH puis on va terminer par la création des organigrammes de système.

4.1. Normalisation des données :

Tout processus de classification commence par une étape d'acquisition des observations qui consiste à déterminer les attributs caractérisant au mieux les objets à classifier.

La normalisation de l'ensemble de données a été effectuée de sorte qu'il serait approprié pour être utilisé par l'algorithme. Les étapes suivantes montrent le processus de normalisation:

Étape 1: Trouver la moyenne de chaque attributs dans l'ensemble de données en utilisant l'équation **[38]**:

$$\text{Mean}_f = \frac{D_{1f} + D_{2f} + \cdots D_{nf}}{n} \qquad (1)$$

où, D_{1f}, D_{2f}…… D_{nf} sont les n valeurs de chaque attributs f

Etape 2 : Calculer l'écart-type de la moyenne calculée en utilisant l'équation **[38]**:

$$\boldsymbol{SD} = \sqrt{\frac{\sum_{i=1}^{n} (\boldsymbol{D_{if} - Mean})^2}{\boldsymbol{n-1}}} \qquad (2)$$

où, D_{1f}, D_{2f}…… D_{nf} sont les n valeurs de chaque attributs f

Étape 3: La valeur normalisée est **[38]** :

$$S_{if} = \frac{D_{if} - Mean_f}{SD_f} \qquad (3)$$

En d'autres termes, pour chaque valeur de caractéristique, nous calculons combien la déviation standard, est-elle loin de la moyenne, et ce résultat devient la nouvelle valeur pour cette fonction. Seules les caractéristiques continues ont été converties mais les caractéristiques nominales ont été conservées comme ils étaient.

Donc, une amélioration proposée ne traite que les caractéristiques nominales. Par exemple X2 est une caractéristique nominale qui est «protocol_type» et sa chaine est la suivante {TCP, UDP, UDP, UDP, RTP, RTP, ICMP, TCP, TCP}, puis nous avons "M = 9 et K = 4" différentes chaînes dans la fonction. Nous pouvons calculer la fréquence relative qui correspond à chaque chaîne dans l'espace réel. Donc, (TCP) = 09/03 = 0,33, (UDP) = 09/03 = 0,33, (RTP) = 09/02 = 0,22, (ICMP) = 09/01 = 0,11. En conséquence, la fonction nominale a été transférée à la représentation numérique suivante (X2 {0.33, 0.33, 0.33, 0.33, 0.22, 0.22, 0.11, 0.33, 0.33}. Sur la base de cette amélioration, nous pourrions étendre les caractéristiques nominales dans l'intervalle [0,1], ce qui est normalisé dans la nature. [37]

4.2. Classification Ascendante Hiérarchique (CAH) : [34]

Le but du module de CAH est d'obtenir une classification automatique de l'ensemble d'individus. Elle commence par déterminer parmi les n individus, quels sont les 2 individus qui se ressemblent le plus par rapport à l'ensemble des p variables spécifiées. Elle va alors regrouper ces 2 individus pour former une classe. Il existe donc à ce niveau (n-1) classes, une étant formée des 2 individus regroupés précédemment, les autres ne contenant qu'un unique individu. Le processus se poursuit en déterminant quelles sont les 2 classes qui se ressemblent le plus, et en les regroupant. Cette opération est répétée jusqu'à l'obtention d'une unique classe regroupant l'ensemble des individus.

Cette procédure est basée sur 2 choix :

- La détermination d'un critère de ressemblance entre les individus. La méthode laisse à l'utilisateur le choix de la dissimilarité.
- La détermination d'une dissimilarité entre classes : procédé appelé un critère d'agrégation. La méthode laisse à l'utilisateur le choix de ce critère.

4.3. Critère d'agrégation : [34]

De nombreux critères d'agrégation ont été proposés les plus connus sont :

Le critère du saut minimal :

La distance entre 2 classes C1 et C2 est définie par la plus courte distance séparant un individu de C1 et un individu de C2.

$$\boldsymbol{D(c_1, c_2) = min\ (\{d(x, y), x\ \in c_1, y\ \in\ c_2} \quad (4)$$

Le critère du saut maximal :

La distance entre 2 classes C1 et C2 est définie par la plus grande distance séparant un individu de C1 et un individu de C2.

$$\boldsymbol{D(c_1, c_2) = max\ (\{d(x, y), x\ \in c_1, y\ \in\ c_2} \quad (5)$$

Le critère de la moyenne :

Ce critère consiste à calculer la distance moyenne entre tous les éléments de C1 et tous les éléments de C2

$$\boldsymbol{D(c_1, c_2) = \frac{1}{n_{c_1}\, n_{c_2}} \sum_{x \in c_1} \sum_{y \in c_2} d(x, y)} \quad (6)$$

Avec :

- nC1 : Le cardinal de C1
- nC2 : Le cardinal de C2

Remarque :

Dans le cas de stratégie du saut minimal, plus une classe contient d'éléments, plus elle est attractive pour des éléments isolés. Au contraire, dans le cas d'une stratégie du saut maximal, plus une classe contient d'éléments, moins elle est attractive pour les éléments isolés.

5. L'algorithme CAH: [35]

Cet algorithme représente les étapes essentielles de la méthode ascendante hiérarchique.

Entrées : I : liste d'individus

Sortie : C : liste de classes initialement vide, une classe est vue comme une liste d'individus

Pas a) calculer les distances d (i, j) pour toute paire (i, j) d'éléments de I

Pas b) trouver i et j ∈ I tels que pour tout i', j' ∈ I : d (i, j) < d (i', j'),

Pas c) pour le couple (i, j)

-former une nouvelle classe c = i ∪ j

-faire I:=I-{i}-{j} et C ={c}

Pas d) si |I| = 0 : STOP

Pas e) calcul des distances entre les classes existantes

i ∈ I et la nouvelle classe c à l'aide de la formule choisie : d (i, c) pour tout i ∈ I

Pas f) faire I := I ∪ C, aller à b)

Pas g) Sélectionner un niveau de coupure de l'arbre pour obtenir le nombre de clusters souhaité.

Initialement, chaque individu forme une classe, on aura ainsi *n* classes. On cherche à réduire le nombre de classes à $nb_{class} < n$, ceci se fait itérativement. À chaque étape, on fusionne deux classes, en réduisant ainsi leur nombre. Les deux classes choisies pour être fusionnées sont celles qui sont les plus « proches », en d'autres termes, celles dont la dissimilarité entre elle est minimale, la première itération a un indice d'agrégation faible, mais celui-ci va croître d'itération en itération.

6. Exemple [36]

Prenons un exemple à quatre observations: a (-1,1), b (0,1), c (3,0) et d (3,-1). On réalise la classification en choisissant comme distance entre individus la distance euclidienne, et comme distance entre classes la distance de Ward. On commence par calculer toutes les distances deux à deux :

	a	b	c	d
a	0	1	4.1	4.5
b		0	3.2	3.6
c			0	1
d				0

Tableau3.1 : Matrice de distances entre les observations deux à deux

Les distances entre a et b d'une part, et c et d d'autre part sont équivalentes et minimales. On peut donc considérer indifféremment la fusion de l'un de ces deux couples. On choisit arbitrairement de fusionner a et b. Il faut maintenant calculer les distances entre la nouvelle classe {a, b} et c, en utilisant La distance de Ward. Le barycentre de la classe {a, b} étant (0.5, 1), on a :

$$\mathrm{D}_w^2\,(\{\mathrm{a},\mathrm{b}\},\mathrm{c}) = \frac{2\times 1}{2+1} d^2 \begin{pmatrix}0.5\\1\end{pmatrix},\begin{pmatrix}3\\0\end{pmatrix} = 4.83$$

$$\Rightarrow D_w\,(\{\mathrm{a},\mathrm{b}\},\mathrm{c}) = 2.2$$

En calculant de la même manière la distance entre {a, b} et d, on met à jour le tableau des distances :

	{a,b}	c	d
{a,b}	0	2.2	2.6
c		0	1
d			0

Tableau3.2 : matrice de distance (2)

On fusionne maintenant les individus c et d, dont la distance est minimale, et on calcule la distance entre la classe {c, d} crée à cette étape et la classe {a, b} :

	{a,b}	{c,d}
{a,b}	0	2.91
{c,d}		0

Tableau3.3 : matrice de distance (3)

A la dernière étape, les deux classes restantes sont fusionnées.

7. Le dendrogramme : [36]

Comme le nom de la méthode (CAH) l'indique, les partitions successivement obtenues sont hiérarchisées : elles sont emboitées les unes dans les autres. De ce fait, il est possible de représenter l'historique des différentes étapes de l'algorithme à l'aide d'une arborescence, aussi appelée dendrogramme. La figure 3.1 montre l'arborescence liée à la CAH réalisée précédemment. En bas de l'arbre se trouvent les individus, jouant le rôle des feuilles. La fusion de deux éléments est représentée par une branche reliant ces deux éléments, dont la hauteur est proportionnelle à la distance entre les deux éléments fusionnés. Ainsi la plus petite branche de l'arbre relie les individus a et b, qui furent les premiers fusionnés. Plus l'on remonte dans l'arbre, plus les classes sont hétérogènes et les branches s'allongent. Dans le cas de la figure 3.1 correspondant à l'exemple considéré, on a choisi de représenter l'arbre avec des hauteurs de branches proportionnelles au pourcentage de perte d'inertie

$$\frac{I_W^{k+1} - I_W^k}{I_T} \qquad (8)$$

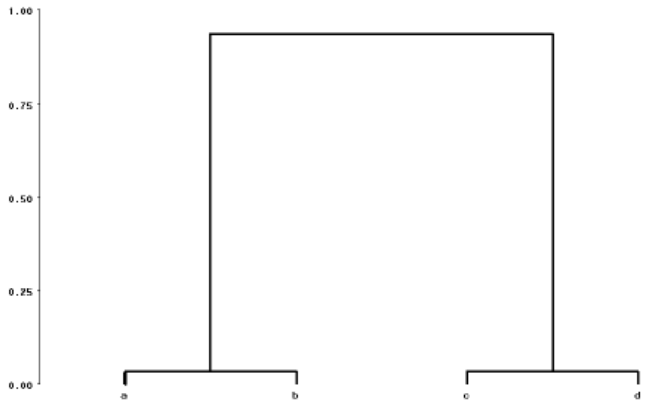

Figure 3.1 : Dendrogramme correspondant à la CAH sur les individus a, b, c et d

Associée à la fusion des deux éléments (lisible à gauche sur l'échelle), plutôt que proportionnelles à la distance entre ces deux éléments.

Cette représentation donne une vision globale de la topologie des observations, et permet d'identifier des classes : on voit ici que les individus a et b forment une classe homogène et bien distincte de l'autre classe regroupant c et d.

Par construction, l'algorithme CAH fournit une classification des données en K classes, pour tout K entre 1 et n. Ainsi, il n'est pas nécessaire de préciser a priori le nombre de classes que l'on souhaite. Ce choix peut être réalisé a posteriori, en considérant le dendrogramme. Les hauteurs des branches étant proportionnelles à la distance entre classes, on peut choisir une classification en "coupant" l'arborescence lorsque les branches sont jugées trop grandes. En effet, une grande branche indique que l'on regroupe des classes qui ne sont pas homogènes.

8. Organigrammes de système :

Notre conception peut être résumée en deux organigrammes essentiels, le premier intitulé normalisation et classification de la base de données KDD. Cette procédure s'exécute une seule fois à la mise en route du système (sauf dans le cas d'une éventuelle mise à jour). Leur résultat est utilisé par le second qui représente l'organigramme du système tout entier dont la tâche est la mesure des distances.

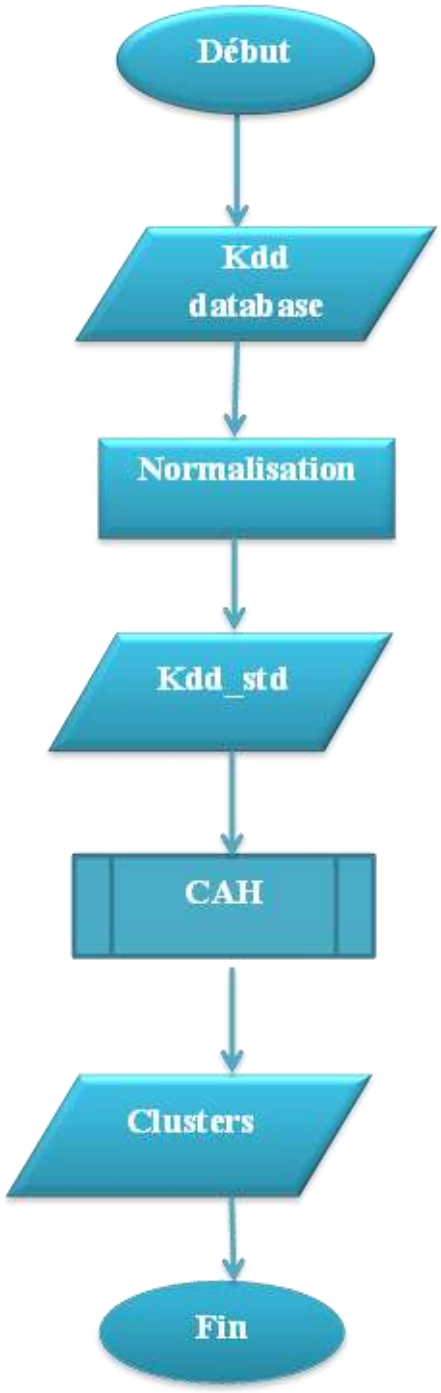

Figure3.2 : Normalisation et classification de la base de données KDD.

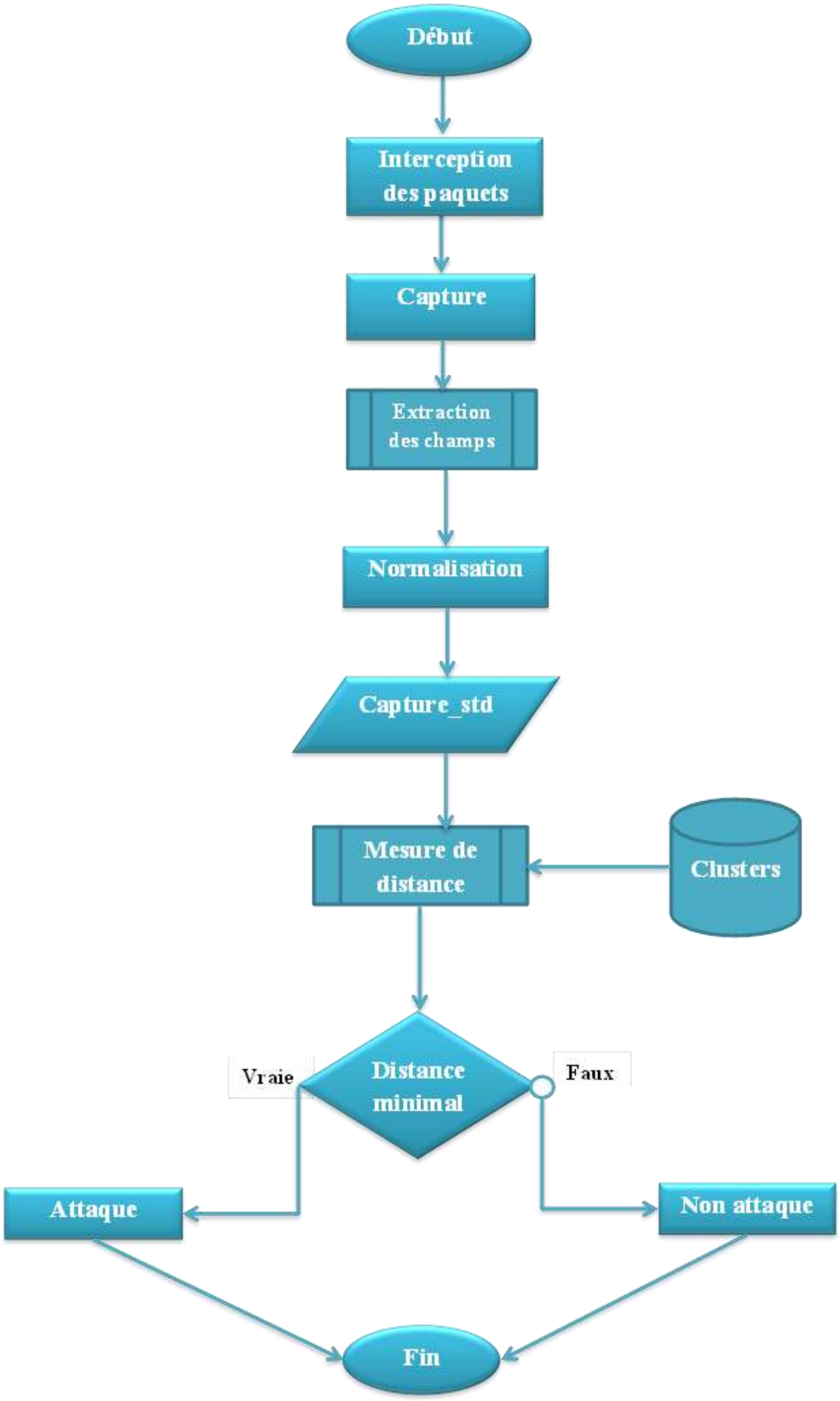

Figure3.3 : organigramme du système

Conclusion

Nous avons décrit dans ce chapitre la démarche que nous avons adoptée pour la mise au point du système de détection d'intrusions en utilisant la CAH ainsi que les différentes étapes de sa conception. Un organigramme récapitulatif est élaboré afin de mieux éclairer le fonctionnement de ce système. Il est à noter que l'utilisation de la CAH dans le domaine de détection d'intrusions est juste à son aube, ce qui justifie la difficulté de concevoir ce type de systèmes. En dépit de ce handicap, nous nous sommes penchés pour fournir un produit pouvant répondre aux aspirations des chercheurs et spécialistes du domaine.

Dans le chapitre suivant, nous allons décrire l'implémentation du système et donner les résultats et discussions des tests effectués afin de dégager des conclusions permettant de justifier notre contribution en la matière.